ELSEVIER

Pattern Recognition Letters 22 (2001) 1033–1042

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Document image template matching based on component block list

Hanchuan Peng [a,b,c,*], Fuhui Long [b], Zheru Chi [b], Wan-Chi Siu [b]

[a] *Department of Radiology, Center for Biomedical Image Computing, School of Medicine, Johns Hopkins University, 601 N. Caroline Street, JHOC 3230, Baltimore, MD 21287, USA*

[b] *Department of Electronic and Information Engineering, Center for Multimedia Signal Processing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*

[c] *Department of Biomedical Engineering, Chien-Shiung Wu Laboratory, Southeast University, Nanjing 210096, China*

Received 21 August 2000; received in revised form 15 February 2001

Abstract

Document image matching is the key technique for document image registration and retrieval. In this paper, a new matching method based on document component block list (CBL) is proposed. A document image is firstly parsed into a number of component blocks that are defined as non-adherent rectangular areas of substantial document contents. Then these blocks are organized as a list, on which several matching operations are defined. The template image that is most similar to the querying document image is selected as the matching result. Our method can effectively make use of the local information of each page component block and the global information of document page layout. We investigate the method with large-scale document template image database. Our method manifests good matching accuracy and good robustness to image distortion, filled-in text, and noises. © 2001 Published by Elsevier Science B.V.

Keywords: Document processing; Image matching; Image registration; Image retrieval; Document database

1. Introduction

Document image registration and retrieval are two important tasks for high quality document image processing systems, which are greatly required in office automation, digital libraries, document databases, etc. The key technique in document image registration and retrieval is document image matching, whose goal is to find out the most similar document image in a registered database for any input document page image. In recent years, many image matching methods have been proposed for specific types of documents. For example, Cesarini et al. (1998) proposed a form-reader system, INFORMys, which used attributed relational graphs to automatically register data forms. Shimotsuji and Asano (1996) presented a cell structure-based two-dimensional hash table to identify different forms. Watanabe et al. (1995) proposed the description of blank form structure that includes the repetitions and positions of cells. Tseng and Chen (1997) presented a form registration method based on three types of line segments. Fan and Chang (1998) calculated the

[*] Corresponding author. Tel.: +1-410-955-7422; fax: +1-410-614-3896.

E-mail addresses: phc@cbmv.jhu.edu (H. Peng), fhlong@eie.polyu.edu.hk (F. Long), enzheru@polyu.edu.hk (Z. Chi).

0167-8655/01/$ - see front matter © 2001 Published by Elsevier Science B.V.
PII: S0167-8655(01)00049-6

line crossing relationship matrix to perform form registration. Luo et al. (1996) proposed an experimental method for identifying content page of documents. Watanabe and Huang (1997) utilized a predefined logical structure to acquire the layout knowledge of business cards. Safari et al. (1997) proposed a projective geometry method to map an input document to the template document.

It is notable that most of the above methods are based on line segments or other local features in the image. Due to the distortion, noises, and the irregular filled-in information on document page images, it is often difficult to find out these local features accurately. Obviously, successful document image matching should combine both global page layout information and local features to produce a reasonable 'representation' of the document image. Such an algorithm should also be robust to image distortion, filled-in text, and noises. For these purposes, here we propose a new document image template matching method based on component block list (CBL) of document image.

This paper is organized as follows. Section 2 briefly explains the data structure in organizing blocks. Section 3 proposes the document image template matching method for general page layout. Section 4 gives a detailed experimental analysis of the proposed algorithm. Discussions and conclusion are presented in Sections 5 and 6, respectively.

2. Component block list

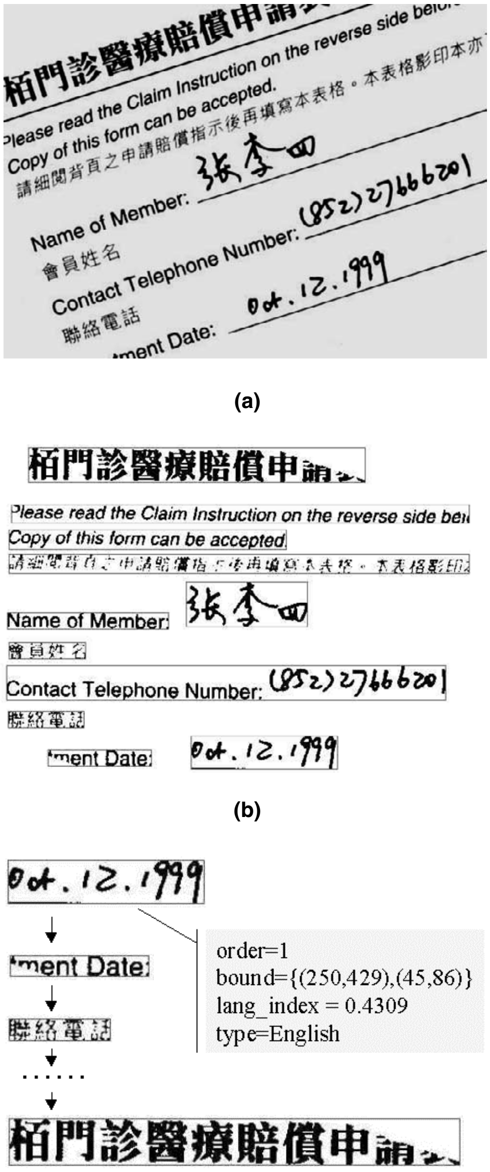
The first step for document image template matching is the extraction of document features, in terms of which the matched template image should be most similar to the input document image. There exist a lot of methods to define and extract document features. Some commonly used features include the line segments, text blocks, labels, etc. Some more abstract features can be attributed relationship graphs, hash tables, projection graphs, etc.

We choose 'component block' as the feature to describe a document page image. In a document image, a component block is defined as an isolated (non-adherent) rectangular area of substantial document contents (texts or graphics). The following two steps are employed to produce component blocks automatically (Peng et al., 2000a):

1. *Preprocessing*: A raw document image, which may contain unknown skewness and noises, is firstly deskewed and then preprocessed to remove unknown distortions and noises. The major foreground of the resulted image is extracted with region-based binarization. Then all long straight lines in the foreground are marked and erased.
2. *Page blocking*: A simple region-growing algorithm is employed to extract all rectangular component blocks. The document image is scanned from bottom to top and from left to right. Each encountered foreground pixel is used as a seed to grow a rectangular component block, which does not contain any more foreground pixel on the outer boundaries. If there is at least one foreground pixel at each of the four boundaries, the block will grow outward one pixel in the direction of that boundary. (A foreground pixel marking procedure is used to prevent duplicated scanning of the foreground pixels, which have already been contained by other component blocks.) This simple algorithm is chosen due to its fast speed because it does not require all pixels in a block are connected with each other.

In our algorithm the component blocks are required to have a minimum size, i.e., too small blocks are taken as noises. An example of the blocked image is shown in Fig. 1(b), which is the result of a grayscale image with large skewness in Fig. 1(a).

The blocked document image is represented by the CBL, which is a one-dimensional data structure of image blocks and is sorted by page blocking order. For example, the CBL of Fig. 1(b) is given in Fig. 1(c), where partial attributes of the first component block are shown. These attributes include blocking order (depending on the position of left-bottom block corner), block boundary (location and size), etc. Documents of different layouts and contents will generally have different CBLs. In this paper, we consider the problem of


Please read the Claim Instruction on the reverse side bel
Copy of this form can be accepted.
Name of Member:
Contact Telephone Number:
ment Date:
order=1
bound={(250,429),(45,86)}
lang_index = 0.4309
type=English

(a)

(b)

(c)

Fig. 1. Partial results in document image blocking and data organizing: (a) the input grayscale image with unknown skewness; (b) results after image binarization, skew correction, and blocking; (c) CBL of the decomposed image and partial attributes of the first block.

image matching for general page layout and only make use of the block location (position of block center) and size attributes.

3. Document image matching algorithm

The CBL of a template page image is named as template block list (TBL). Given a template image database, the problem of document page image matching turns to be finding out the most similar TBL in the database to the CBL of an input page image, whose blocks contain various deformations. For document image of general page layout, we propose a new method referred to here as CBL-based matching algorithm (CBL-MA). The method can be described with the following pseudocode:

```
**Procedure** CBL-MA;
{Input: a CBL for the input document image, a handle to a template image database of $K$ TBLs}
{Output: the TBL with the minimum distance $D$ to CBL}
{Preprocessing: **for** $k = 1,\ldots,K$, **do begin** sort the $k$th TBL by block size (from small to large); **end**.}
{Note: the preprocessing is not a part of this CBL-MA and needs to be done only once beforehand}
**begin**
    sort the CBL by block size (from small to large);
    **for** $k = 1,\ldots,K$, **do begin**
        compute $D_k$, which is the distance between CBL and the $k$th TBL;
    **end**;
    select the TBL with the minimum $D_k$ as output;
**end.**
```

Denote the ranges of CBL and TBL block indexes as $R^{\text{CBL}} = [I^{\text{CBL}}_{\min}, I^{\text{CBL}}_{\max}]$ and $R^{\text{TBL}} = [I^{\text{TBL}}_{\min}, I^{\text{TBL}}_{\max}]$, respectively. The distance D between CBL and TBL is computed as follows:

1. *Size matching*: For each block in TBL, find the most similar block in CBL according to size. Denote the size of the ith block in TBL as S_i^{TBL}, and the size of jth block in the CBL as S_j^{CBL}. This step is to find the CBL block with the following index I_S:

$$I_S = \arg\min_{j\in R^{\text{CBL}}}\left\{d_S\left(S_j^{\text{CBL}}, S_i^{\text{TBL}}\right)\right\}, \tag{1}$$

where the matching degree of block sizes, d_S, is defined as the absolute value of block area difference

$$d_S\left(S_j^{\mathrm{CBL}}, S_i^{\mathrm{TBL}}\right) = d_S\left(A_j^{\mathrm{CBL}}, A_i^{\mathrm{TBL}}\right) = \left|A_j^{\mathrm{CBL}} - A_i^{\mathrm{TBL}}\right|, \tag{2}$$

where A_i^{TBL} and A_j^{CBL} are areas of the ith TBL block and the jth CBL block, respectively.

2. *Location matching*: In CBL, search the neighbors (within a given neighborhood) of the found CBL blocks for the most similar page blocks in location. Denote the center of the ith block in TBL as C_i^{TBL}, and the center of the jth block in the CBL as C_j^{CBL}, this step is to find the CBL block with the following index I_C:

$$I_C = \mathop{\arg\min}_{j \in [I_S - T_C, I_S + T_C] \cap R^{\mathrm{CBL}}} \left\{ d_C\left(C_j^{\mathrm{CBL}}, C_i^{\mathrm{TBL}}\right) \right\}, \tag{3}$$

where T_C is a predefined neighborhood threshold, and the matching degree of block centers, d_C, is defined as the displacement of block centers (in norm-2)

$$d_C\left(C_j^{\mathrm{CBL}}, C_i^{\mathrm{TBL}}\right) = \left\|C_j^{\mathrm{CBL}} - C_i^{\mathrm{TBL}}\right\|. \tag{4}$$

3. *Distance calculation*: The ‘distance’ between an input document image and a template image is defined as the sum of d_C over R^{TBL}:

$$D = \sum_{i \in R^{\mathrm{TBL}}} d_C\left(C_{I_C}^{\mathrm{CBL}}, C_i^{\mathrm{TBL}}\right). \tag{5}$$

CBL-MA calculates distances between the input document image and all templates in the database and selects the template with the minimum distance (maximum similarity) as the matching result. The mechanism of the above algorithm can be illustrated in Fig. 2: the block template B^{T} first best matches block A (size matching) and then is adjusted to best match block B (location matching), though B is severely deformed from B^{T}.

From a viewpoint of data retrieval, the general page layout matching is very difficult because it turns to be a high-dimensional indexing problem, where efficient methods for both high-dimension reduction and multi-dimensional indexing are needed (if we take each block in TBL/CBL as a one-dimensional feature). CBL-MA gives a different, and simpler, solution by sequentially performing one-dimensional sorting/matching/searching for block features and TBL/CBL distances. These operations can be implemented with high efficiency. For instance, we can use B-tree to organize CBL and use binary search to find out the I_S for each TBL block. One outstanding advantage is that both the local information of blocks and the global information of document layout can be integrated.

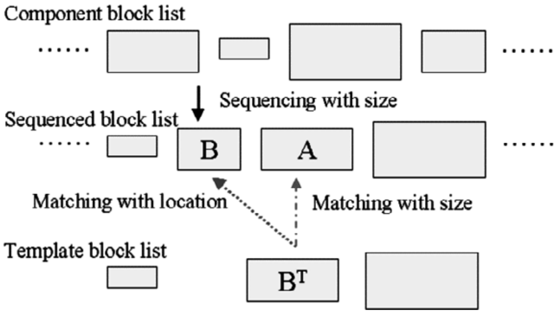


Fig. 2. Mechanism of CBL-MA.

4. Experimental results

4.1. Experimental database

We use a large-scale document image database of 1350 document templates. All template images are firstly normalized to 1024 × 768 in size and then blocked and stored in the database as TBL. Each TBL contains about 50 component blocks. The minimum size of a template component block is 8 pixels in height and 16 pixels in width. The minimum space between template component blocks is 8 pixels. For the sake of experimental investigation, we define (without a priori knowledge) four independent (non-overlapping) subsets of the database, which contain 50, 100, 200, and 500 templates, respectively. Denote these four subsets as *Set-A*, *Set-B*, *Set-C*, and *Set-D*, sequentially. Denote the whole database with all 1350 templates as *Set-E*.

We design experiments to examine the performance of CBL-MA for different degrees of document image deformations and different template set size. Note that in many office automation applications (e.g., document image registration), generally a document template set contains less than 100 document templates (however for digital library applications, e.g., image retrieval, where the template set size can be huge, see the discussion in Section 5). Hence, for image deformation, including global deformation (i.e., block detection errors, including block misdetection and misaddition) and local deformation (i.e., block disfiguration, including block location variation, block size variation, block rotation, etc.), we employ *Set-B* to examine the performance of CBL-MA. For the influence of the template set size on the matching accuracy, we use all the five template sets.

We use computer to automatically generate random test images and model the various image deformations and image blocking errors. The procedure consists of five independent steps (for five different cases), which are block misdetection (the block does not appear in the CBL), block misaddition (additional blocks are detected due to noises, large text spaces, line drop of graphics, etc.), block size variation (for both block width and height, due to the document irregular texts and noises), block location variation (due to the irregular block information), and block rotation (note that block rotation also result in block size variation, however the block width and height vary in different degrees). The corresponding parameters are: the block misdetection rate P_{m}, block misaddition rate P_{a}, block size deformation rate P_{s}, and size deformation scale factor S_{s}, block location displacement rate P_{d} and displacement scale factor S_{d}, rotation probability P_{r} and rotation angle D_{r}.

Fig. 3(b) is a generated image from the template shown in Fig. 3(a). The parameters are $\{P_{\mathrm{m}} = 0.2,\ P_{\mathrm{a}} = 0.2,\ P_{\mathrm{s}} = 0.2,\ S_{\mathrm{s}} = 0.2,\ P_{\mathrm{d}} = 0.5,\ S_{\mathrm{d}} = 0.5,\ P_{\mathrm{r}} = 0.5,\ D_{\mathrm{r}} = 15^\circ\}$. Obviously, these two images are significantly different. Actually many computer-generated images in our experiments have larger variation to the original template images than samples scanned from physical document pages.

For the following reported experimental results, the neighborhood threshold in location matching, T_C, is 3.

4.2. Performance for global deformation (block detection errors)

Set-B is used for investigation. For each kind of parameters settings, 400 test images (4 deformed images are generated for one template) are generated to examine the influence of P_{m} and P_{a} on the matching accuracy (correct classification rate) r_{c} of CBL-MA. From the results listed in Table 1, we find:

1. CBL-MA can perform well ($r_{\mathrm{c}} > 85\%$) even when 50% blocks in the block list are lost or wrongly added (see the column of $P_{\mathrm{m}} = 0.5$ and $P_{\mathrm{a}} = 0.5$). Even when 80% blocks are lost or added, this algorithm can still produce matching accuracy nearly 60% (see the column of $P_{\mathrm{m}} = 0.8$ and $P_{\mathrm{a}} = 0.8$).
2. CBL-MA is more insensitive to block misaddition than to block misdetection. This is reasonable because when additional blocks are wrongly put into CBL, the original blocks still play, though weaker, roles. On the contrary, the lost information due to block misdetection is non-recoverable.

Notice that block misdetection/misaddition is a kind of global deformation because the whole CBL is changed. From this experiment we see CBL-MA is robust to block detection errors.

4.3. Performance for local deformation (block disfiguration)

Set-B is used for investigation. For each kind of parameters settings, 400 test images are generated to examine the influence of three types of block disfiguration (size variation, location variation, and rotation) on the matching accuracy r_{c} of CBL-MA.

For block size variation, the influence of parameters P_{s} and S_{s} (here we set the same scale factor for both block width and height) on r_{c} is given in Table 2. When blocks expand or shrink greatly, CBL-MA can keep r_{c} above 90% (the fourth row of Table 2). At the same time, even when all blocks have size variations ($P_{\mathrm{s}} = 1.0$), our

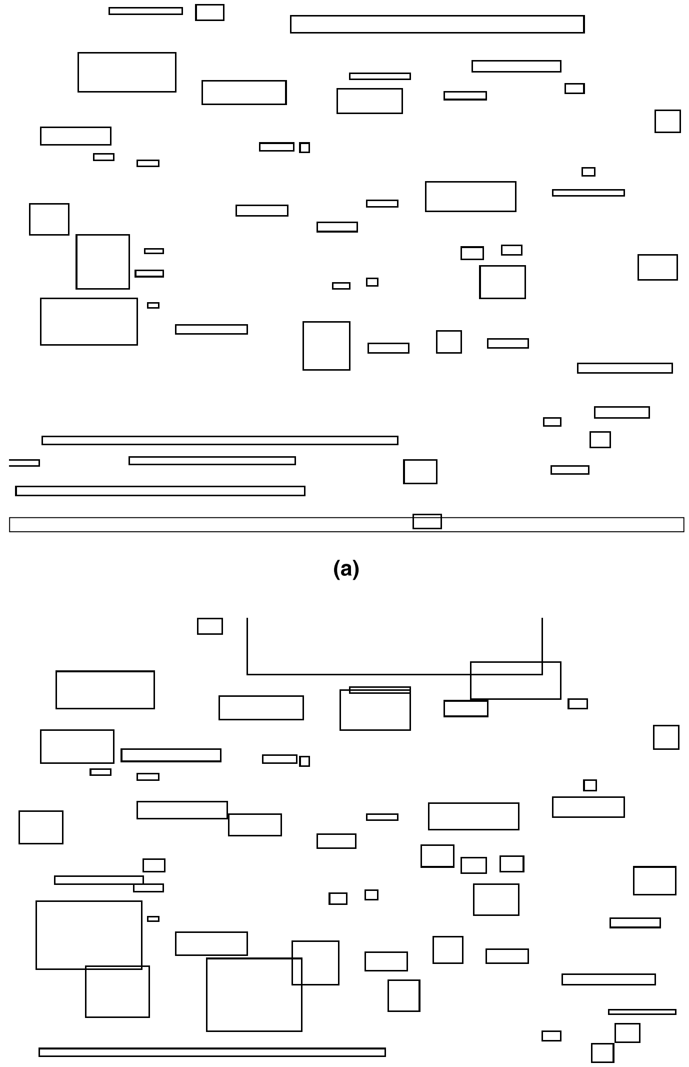


Fig. 3. (a) A template image example; (b) one deformation image of (a).

Table 1
Matching accuracy for block misdetection and misaddition ($P_{\mathrm{s}} = 0.2$, $S_{\mathrm{s}} = 0.2$, $P_{\mathrm{d}} = 0.5$, $S_{\mathrm{d}} = 0.5$, $P_{\mathrm{r}} = 0.5$, $D_{\mathrm{r}} = 15°$)

P_{m} ($P_{\mathrm{a}} = 0.1$)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
r_{c} (%)	97.5	94.75	93.5	90.0	87.0	82.5	68.25	58.75	46.75
P_{a} ($P_{\mathrm{m}} = 0.1$)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
r_{c} (%)	97.5	94.75	93.0	85.25	85.25	79.25	75.5	71.25	70.0

Table 2
Matching accuracy for block size variation ($P_a = 0.1$, $P_m = 0.1$, $P_d = 0.5$, $S_d = 0.5$, $P_r = 0.5$, $D_r = 15°$)

P_s ($S_s = 0.2$)	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
r_c (%)	97.5	95.75	96.25	95.5	95.25	95.25	95.25	95.0	95.0
S_s ($P_s = 0.2$)	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
r_c (%)	97.5	97.25	96.5	96.5	96.0	94.0	93.5	92.75	92.25

algorithm can produce a very high matching accuracy of 95%. Note that the latter corresponds to many office automation applications, where S_s is not very large, however, most blocks are subject to some degree of size variation, i.e., P_s is close to 1.

For block location displacement, the influence of parameters P_d and S_d on r_c is given in Table 3. Evidently CBL-MA is robust to block location variation (r_c always larger than 95%). (In Tables 2 and 3, the little fluctuation of some data, which should monotonously decrease, is due to the occasional instability of computer-generated test images.)

For block rotation, the influence of parameters P_r and D_r on r_c is given in Table 4. For both cases $\{D_r = 15°, P_r$ varies from 0.2 to 1.0$\}$ and $\{P_r = 0.5, D_r$ varies from 5° to 45°$\}$, CBL-MA produces satisfying classification, even when the test images contain strong deformation, e.g., 50% component blocks have at most 45° rotation, or all component blocks have at most 15° rotation. Notice that block rotation will directly lead to the significant change of block sizes.

The above three experiments show that CBL-MA is robust to local deformation of document image.

4.4. Performance for different template set size

Here under a general setting of parameters $\{P_a = 0.2,\ P_m = 0.2,\ P_s = 0.2,\ S_s = 0.2,\ P_d = 0.5,\ S_d = 0.5,\ P_r = 0.5,\ D_r = 15°\}$, we examine the influence of the template image set size on the matching accuracy. All the five template sets are used. For each template set, we independently generate at least 2000 images for testing. The results are listed in Table 5. It is clear that even when the template set size grows to 500, the matching accuracy is satisfying (>80%). For the 1350-template set (*Set-E*), r_c is still around 70%.

Table 3
Matching accuracy for block location variation ($P_a = 0.1$, $P_m = 0.1$, $P_s = 0.2$, $S_s = 0.2$, $P_r = 0.5$, $D_r = 15°$)

P_d ($S_d = 0.5$)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
r_c (%)	99.5	99.25	99.0	98.25	98.0	97.5	97.5	97.25	97.0
S_d ($P_d = 0.5$)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
r_c (%)	98.5	98.25	98.0	97.5	98.0	97.5	96.75	96.5	96.75

Table 4
Matching accuracy for block rotation ($P_a = 0.1$, $P_m = 0.1$, $P_s = 0.2$, $S_s = 0.2$, $P_d = 0.5$, $S_d = 0.5$)

P_r ($D_r = 15°$)	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
r_c (%)	100.0	100.0	99.5	97.5	95.25	87.25	79.75	63.5	57.0
D_r ($P_r = 0.5$)	5°	10°	15°	20°	25°	30°	35°	40°	45°
r_c (%)	100.0	99.5	97.5	95.25	93.0	92.25	87.5	87.0	82.25

Table 5
Matching accuracy for different template set size ($P_a = 0.2$, $P_m = 0.2$, $P_s = 0.2$, $S_s = 0.2$, $P_d = 0.5$, $S_d = 0.5$, $P_r = 0.5$, $D_r = 15°$)

Template set size	50	100	200	500	1350
Test image numbers	2000	2000	2000	2000	2700
r_c (%)	92.30	91.45	87.40	83.60	70.00

5. Discussion

5.1. Comparison to other algorithms

Different from those local features-based document image matching algorithms, CBL-MA successfully integrates the local and global information of document image. It is noticed that the failure in detecting local features (e.g., line-segments) usually immediately results in bad performance of several other algorithms. In contrast, our method is robust to filled-in contents, image distortion, and noises. For example, Fan and Chang (1998) used the line-based features to describe a form, where the broken lines or the text-line-adherence will lead to inaccurate line crossing matrix and poor performance of the whole algorithm. Unfortunately, in real environments, these two bad cases are often aroused by improper document imaging resolution or conditions (e.g., too low or too high resolution, distortion of the documents, etc.). These cases will also do harm to the Tseng and Chen (1997) line-segment-based form registration method. In the Safari et al. (1997) approach, it was claimed that key points selection is critical and a certain number (e.g., 5) of the key points are necessary to guarantee the performance of the algorithm. However, our method presents robustness to local feature detection by making use of a good data structure to organize component block location and size attributes and the global page layout. Our experiments demonstrate that even when the block information is partially lost or inaccurate (especially when the page blocking algorithm is very coarse), there is no significant performance reduction of CBL-MA.

5.2. Further analysis of our method

The computational complexity of our method is analyzed as follows. Denote n as the number of blocks in a CBL, m as the number of blocks in a TBL, K as the number of template images in a document image database. When we use quicksort and binary search algorithms, the typical computational complexity is $\mathrm{O}(n \log n)$ for CBL sorting, $\mathrm{O}(Km \log n)$ for CBL/TBL size matching, $\mathrm{O}(Km(2T_C + 1))$ for CBL/TBL location matching, $\mathrm{O}(K)$ for CBL/TBL distance calculation, and $\mathrm{O}(\log K)$ for finding the minimum distance. The total computational complexity of CBL-MA is $\mathrm{O}(n \log n) + \mathrm{O}(Km \log n) + \mathrm{O}(Km(2T_C + 1)) + \mathrm{O}(K) + \mathrm{O}(\log K) = \mathrm{O}((Km + n) \log n)$. We see that the algorithm has the linear time complexity with the template set size and TBL block number (note that all TBLs will be sorted in preprocessing), respectively. Clearly, for a middle-sized template document set, the computational burden is not heavy.

The component block-based document image matching is closely related to high-dimensional data organization and retrieval. A detailed discussion from viewpoint of database is beyond this paper. However, with CBL-MA we can simplify this problem to sorting/searching one-dimensional data. For a very large template image set or TBL with a large number of blocks, because the high-dimensional indexing-based approach is difficult, we consider the following simpler solution of 'hierarchical' CBL-MA:

1. When the template set size is very large, we can pre-cluster template images into a certain number of template sets with much smaller sizes. Each small template set is represented by its cluster center. For any querying document image, we first match it to these clustering centers and decide which small template set it belongs to. Then find the best matching template of the input document image in the found small set. This method can reduce a large amount of computation (depending on the number of small sets), though the matching error may rise (depending on the clustering algorithm). It is noticed that CBL-MA itself can also be applied to clustering template images.
2. When the block number in TBL is large, we can pre-grade the blocks in TBL according to their importance (e.g., the most important blocks are the N largest blocks in TBL, where N is a predefined number). For any querying document image, we first match the N most important blocks in CBL with N most important blocks in each TBL and find out the M (a predefined number) best matching TBLs. Then match the $2N$ most important blocks in CBL with the $2N$ most

important blocks in each of the M TBLs and find out the $M/2$ best matching TBLs. Repeat the procedure until the best matching TBL is found. Performance of the algorithm depends on the choices of the importance measurement of block, and parameters N, M. This solution is actually a dimension reduction approach for the high-dimensional data.

It is necessary to clarify the following three points:

1. Our method does not care about the content of each component block. This is quite reasonable for some applications, especially automatic data form reading, where an input document image is required to be automatically mapped to one of pre-stored database tables. Because the data in each document field vary from image to image (even when the images are of the same type, i.e., they should be mapped to the same database table), CBL-MA can be used to find out the correct database table of the input document image, and then the logical structure of the found database table can be used to annotate the component blocks in the input image. In one of our recent software packages, CBL-MA has been applied to automatic data reading of industrial column forms (Peng et al., 2000b,c).
2. In CBL-MA it is not encouraged to exchange the order of size operations and location operations. We find that in many cases the size-first sorting has less block order derangement (due to the detection error of single block) than the location-first sorting. Consequently, the size-first matching requires a significantly smaller neighborhood threshold than the location-first matching. In another words, to our experiences, the size-first sorting/matching has a larger possibility to produce better performance (in terms of matching accuracy and computation time) than location-first sorting/matching.
3. The quality of the template image set is important. If some TBLs themselves can arouse confusion, they should be classified as just one TBL. If these template images have to be distinguished, the only solution is introducing other features of the documents.

5.3. Applications

CBL-MA can serve as one of the key algorithms of many document image registration and retrieval applications. Concrete examples include form data auto-reading, automatic document categorization for digital library, document data sharing in video-conferencing, etc.

6. Conclusion

In this paper, a new method of document image matching is proposed based on the component block list (CBL). The method can effectively integrate the local and global features of a document image for image matching. The computer simulations demonstrate that our method is robust to image deformation, filled-in information and noise: for middle-sized template image set of about 100 templates, the proposed method can attain matching accuracy above 90%, and even when the template set size goes up to 500, our method can still attain accuracy above 83%. (Some more materials of this paper are available at http://pandora.cbmv.jhu.edu/~phc/.)

Acknowledgements

This paper is partially supported by a grant from the Hong Kong Polytechnic University (No. A042, which is under the ASD of Multimedia Media Processing) and partially supported by a Chinese NSF Key Project (No. 69831010) in Chien-Shiung Wu Laboratory, Department of Biomedical Engineering, Southeast University, Nanjing, China. The authors would like to thank the anonymous reviewers' detailed and constructive comments to improve this paper.

References

Cesarini, F., Gori, M., Marinai, S., Soda, G., 1998. INFORMys: a flexible invoice-like form-reader system. IEEE Trans. Pattern Anal. Machine Intell. 20 (7), 710–745.

Fan, K., Chang, M., 1998. Form document identification using line structure based features. In: Proc. 14th Internat. Conf. on Pattern Recognition, Vol. 2, pp. 1098–1100.

Luo, Q., Watanabe, T., Makayama, T., 1996. Identifying contents page of documents. In: Proc. 13th Internat. Conf. on Pattern Recognition, Vol. 3, pp. 696–700.

Peng, H., Chi, Z., Siu, W., Feng, D., 2000a. PageX: an integrated document processing software for digital libraries. In: Proc. 2000 Internat. Workshop on Multimedia Data Storage, Retrieval, Integration, and Applications, Hong Kong, pp. 203–207.

Peng, H., Long, F., Feng, D., Siu, W., 2000b. A heterogeneous document database model based on component blocks. CMSP Technical Reports, Hong Kong Polytechnic University.

Peng, H., Long, F., Siu, W., Chi, Z., Feng, D., 2000c. Document image matching based on component blocks. In: Proc. 2000 Internat. Conf. on Image Processing, Vancouver, Canada, pp. 601–604.

Safari, R., Narasimhamurthi, N., Shridhar, M., Ahmadi, M., 1997. Document registration using projective geometry. IEEE Trans. Image Process. 6 (9), 1337–1341.

Shimotsuji, S., Asano, M., 1996. Form identification based on cell structure. In: Proc. 13th Internat. Conf. on Pattern Recognition, Vol. 3, pp. 793–797.

Tseng, L., Chen, R., 1997. The recognition of form documents based on three types of line segments. In: Proc. 4th Internat. Conf. on Document Analysis and Recognition, Vol. 1, pp. 71–75.

Watanabe, T., Huang, X., 1997. Automatic acquisition of layout knowledge for understanding business cards. In: Proc. 4th Internat. Conf. on Document Analysis and Recognition, Vol. 1, pp. 216–220.

Watanabe, T., Luo, Q., Sugie, N., 1995. Layout recognition of multi-kinds of table form documents. IEEE Trans. Pattern Anal. Machine Intell. 17 (4), 432–445.